

Streamline integration as a method for two-dimensional elliptic grid generation

M. Wiesenberger, M. Held

*Institute for Ion Physics and Applied Physics, Universität Innsbruck, Technikerstraße
25, A-6020 Innsbruck, Austria*

L. Einkemmer

Numerical Analysis group, Universität Innsbruck, A-6020 Innsbruck, Austria

Abstract

We propose a new numerical algorithm to construct a structured numerical grid of a doubly connected domain that is bounded by the contour lines of a given function. It is based on the integration of the streamlines of the two vector fields that form the basis of the coordinate system. These vector fields are either built directly from the given function or from the solution of a suitably chosen elliptic equation (which can be solved once an initial grid has been constructed). We are able to construct conformal, orthogonal and curvilinear coordinates. The method is parallelizable and the metric elements can be computed with high accuracy. Furthermore, it is easy to implement as only the integration of well-behaved ordinary differential equations and the inversion of a linear elliptic equation are required. All our grids are orthogonal to the boundary of the domain, which is the major advantage over previously suggested grids. We assess the quality of our grids with the solution of an elliptic equation and the distribution of cell sizes. We find that simple flux aligned orthogonal grids are suitable for the solution of flux aligned problems, which is to be expected, but that they exhibit very large ratios of maximal to minimal cell size. In the conformal grid the aspect ratio of the cells is constant by construction. However, the variation in cell size is large and the errors of the elliptic equation are high. The adapted grid and the grid with monitor metric yield smaller variation in size and smaller

Email address: Matthias.Wiesenberger@uibk.ac.at (M. Wiesenberger)

errors, where the monitor grid overall has the smallest ratios of maximal to minimal cell size. The errors and cell sizes are competitive with a previously suggested near conformal grid.

Keywords: elliptic grid generation, discontinuous Galerkin, conformal grid generation, orthogonal grids

1. Introduction

Tokamaks are magnetic fusion devices that use nearly axisymmetric magnetic fields to confine a plasma [1]. In a two-dimensional plane of fixed toroidal angle the magnetic field-lines, in an idealized case, lie within so-called flux-surfaces. These surfaces are in fact the isosurfaces of the poloidal magnetic flux $\psi(x, y)$ described by the Grad-Shafranov equation. A typical numerical modelling scenario involves the study of plasma dynamics between two flux-surfaces ψ_0 and ψ_1 . Structured and unstructured numerical grids have been proposed for the numerical discretization of this region [2, 3, 4, 5, 6, 7, 8, 9]. In this work we only discuss structured grids, i.e. grids that are generated by the (numerical) coordinate transformation of a rectangular computational domain to the physical domain [10, 11, 12]. We choose this approach because derivatives are easier to implement and cheaper to compute on a tensor product grid (the computational domain is a rectangle) compared to unstructured grids. Generally, the same numerical techniques as in a Cartesian coordinate system can be used. On the other hand, the physical equations need to be transformed onto the computational domain and metric elements might appear explicitly in the equations. From a numerical viewpoint, if u_i, v_i are the discrete point coordinates in computational space, we need to construct the inverse coordinate transformation $x_i = x(u_i, v_i)$, $y_i = y(u_i, v_i)$, as well as its derivatives $\partial u / \partial x|_{(u_i, v_i)}$, $\partial v / \partial x|_{(u_i, v_i)}$, $\partial u / \partial y|_{(u_i, v_i)}$ and $\partial v / \partial y|_{(u_i, v_i)}$. The latter is required in order to transform scalar fields and tensors to the curvilinear coordinate system.

A popular choice for tokamak magnetic fields are so-called flux or magnetic coordinates [13], in which the magnetic flux ψ or a function of it is the first coordinate. One particular class of such coordinate systems is known as PEST coordinates [5]. In this case, the geometric toroidal angle φ is the third coordinate, while a poloidal angle-like coordinate is constructed implicitly by choosing the volume form of the coordinate transformation. Other

choices of flux coordinates exist, where the geometric toroidal angle φ differs from the toroidal flux angle φ_f . For example, Boozer [3, 4] and Hamada [2] coordinates.

In general a flux aligned coordinate system is not orthogonal. In actual turbulence simulations this is a major drawback. Reference [10] pointed out that in order to implement Neumann boundary conditions it is desirable for the grid to be orthogonal or near-orthogonal to the boundary. Although it is possible to represent Neumann boundary conditions in a curvilinear grid, the accuracy of the discretization deteriorates and the implementation is more involved than in a boundary-orthogonal grid. Another general desirable grid quality is a homogeneous distribution of cells across the domain. This condition is important in advection schemes, where too small cells deteriorate the CFL condition and too large cells deteriorate the accuracy. In this paper we will show how to construct grids that are orthogonal to the boundary and distribute cells evenly across the domain.

In the literature on grid generation it has been established that elliptic grids are the ones that best fulfill these qualities [10, 11, 12]. These are grids generated by a coordinate transformation that fulfills an elliptic equation. However, the generation of these grids based on the algorithm proposed by Thompson, Thames and Mastin (TTM) is numerically involved as it requires the solution of the inverted Beltrami equation [14, 12].

One special class of elliptic grids is generated by conformal mappings. The coordinates obey the Cauchy–Riemann differential equations

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (1)$$

They are orthogonal, smooth, and preserve the aspect ratio of the cells from the computational into the physical space. There are many methods to numerically construct a conformal mapping [15], e.g. boundary integral element methods, however their implementation can be tricky. Among further improvements to the original TTM method are grid adaption methods and the monitor metric approach [16, 17, 12]. Both of these approaches modify the elliptic equation used to construct the coordinate system to a more suitable form. By choosing specific forms the distribution of grid cells is controlled, which makes the coordinate transformation more flexible than conformal mappings.

The previously discussed flux aligned grids are commonly constructed by integrating the streamlines of the vector field tangential to the isosurfaces of

ψ [13, 18]. Reference [9] constructed a near-conformal coordinate system that is aligned to the magnetic flux-surfaces in this way. The coordinates are near-conformal in the sense that the grid-deformation is small. Unfortunately, the coordinate lines are not orthogonal.

The main reason to generate an elliptic grid via the solution of the inverted Beltrami equation is that the (not yet constructed) numerical grid is needed in order to discretize and solve the elliptic equation directly. If the boundary lines are e.g. given by a parametric representation, this conundrum cannot be easily avoided. However, as already pointed out the existence of the function $\psi(x, y)$ that implicitly determines the boundary of the domain allows us to construct a numerical grid easily.

We therefore propose to construct an elliptic grid in three steps. First, a flux aligned grid is constructed by one of the methods that have been proposed in the literature. We choose the angle-like coordinate to be the arc length of the lines of constant ψ . Then in a second step a suitably chosen elliptic equation is discretized and solved on this grid. In this step we investigate the simple Laplace equation, the adapted equation and the elliptic equation with monitor metric suggested by References [12, 16, 17]. We use discontinuous Galerkin methods to discretize the system in the coordinate frame constructed in the first step. Finally, we treat the solution of the elliptic equation as the new flux function and use it to again construct a flux aligned grid via streamline integration.

In Section 2 we derive our algorithm by using methods from differential geometry. We explain the approach of streamline integration by constructing an orthogonal flux aligned grid. Here, we rederive the relevant equations in terms of 1-forms and partial differentials [19] as opposed to the formulation in terms of covariant and contravariant unit vectors [13]. We then show how to construct a conformal grid, the adapted grid and the grid using a monitor metric. In Section 3 we show that with our algorithm the coordinate map and its derivatives can be computed efficiently and up to machine precision. Furthermore, we plot example grids for a domain close to the separatrix, which are thus suitable for edge turbulence simulations. We compare the grids to flux aligned grids and assess the quality of our grids with two different solutions of an elliptic equation and the computation of maximal and minimal cell sizes.

2. High precision elliptic grid generation

Given is a function $\psi(x, y)$ in Cartesian coordinates. We want to construct a grid on the region bounded by the two lines $\psi(x, y) = \psi_0$ and $\psi(x, y) = \psi_1$ with $\psi_0 \neq \psi_1$. The derivative of the function ψ may not vanish within this region and on the boundary and we further assume that the region is topologically a ring. Note here that this excludes the description of domains with an X-point or O-point. The numerical grid is described by a mapping of the discretization of the rectangular computational domain $(u, v) \in [0, u_1] \times [0, 2\pi]$ to the physical domain (x, y) . u_1 is an unknown, which the grid generation process has to provide. Together with this mapping described by $x_i = x(u_i, v_i)$, $y_i = y(u_i, v_i)$ we construct the derivatives $u_x(u_i, v_i)$, $v_x(u_i, v_i)$, $u_y(u_i, v_i)$, $v_y(u_i, v_i)$ in order to transform tensors to the curvilinear coordinate system. Here and in the following we use the notation $u_x \equiv \frac{\partial u}{\partial x}$. The index i is given by $i = 0, 1, \dots, N_u N_v$, where N_u and N_v are the number of points in u and v respectively. The discretization of the computational domain need not be equidistant. We will for example use the Gauss-Legendre nodes needed for a discontinuous Galerkin discretization.

We first introduce some helpful quantities in Section 2.1. In Section 2.2 we review the construction of orthogonal coordinates via the integration of streamlines, that is ordinary differential equations. In the next step in Section 2.3 we discretize and solve the Laplace equation in a flux aligned coordinate system with local discontinuous Galerkin (dG) methods. The solution is then used to replace ψ as a flux function in Section 2.4, where we transform the Cauchy-Riemann equations (1) in order to construct conformal coordinates. In Section 2.5 we extend our algorithm to allow grid adaption, before we introduce the monitor metric method in Section 2.6. Finally, we generalize our algorithm in Section 2.7.

2.1. Preliminaries

In a curvilinear coordinate system (u, v) the metric tensor and its inverse take the form

$$\mathbf{g}(u, v) = \begin{pmatrix} g_{uu} & g_{uv} \\ g_{uv} & g_{vv} \end{pmatrix} \quad \mathbf{g}^{-1}(u, v) = \begin{pmatrix} g^{uu} & g^{uv} \\ g^{uv} & g^{vv} \end{pmatrix} \quad (2)$$

We denote $g = \det \mathbf{g} = g_{uu}g_{vv} - g_{uv}^2$. The elements of the inverse metric tensor are given by

$$g^{uu} = u_x^2 + u_y^2 \quad (3a)$$

$$g^{uv} = u_x v_x + u_y v_y \quad (3b)$$

$$g^{vv} = v_x^2 + v_y^2 \quad (3c)$$

since $g^{xx} = g^{yy} = 1$ and $g^{xy} = 0$. Given $u(x, y)$, $v(x, y)$ and its inverse $x(u, v)$, $y(u, v)$ recall that the derivatives are related by

$$\begin{pmatrix} x_u & x_v \\ y_u & y_v \end{pmatrix} \Big|_{u(x,y), v(x,y)} = \frac{1}{u_x v_y - u_y v_x} \begin{pmatrix} v_y & -u_y \\ -v_x & u_x \end{pmatrix} \Big|_{x,y} \quad (4)$$

The volume form is given by the determinant via

$$\mathcal{V}^2 := dx \wedge dy = (x_u y_v - y_u x_v) du \wedge dv = \frac{du \wedge dv}{u_x v_y - u_y v_x} \equiv \sqrt{g} du \wedge dv \quad (5)$$

The elements of the divergence and gradient operators in an arbitrary coordinate system read

$$(\nabla f)^i = g^{ij} \partial_j f \quad (6a)$$

$$\nabla \cdot \mathbf{A} = \frac{1}{\sqrt{g}} \partial_i (\sqrt{g} A^i), \quad (6b)$$

where we sum over repeated indices $i, j \in \{u, v\}$ and define $\partial_u \equiv \partial/\partial u$, $\partial_v \equiv \partial/\partial v$. Finally, we introduce the geometrical poloidal angle

$$\theta(x, y) = \begin{cases} + \arccos \left(\frac{x-x_0}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \right) & \text{for } y \geq y_0 \\ - \arccos \left(\frac{x-x_0}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \right) & \text{for } y < y_0 \end{cases} \quad (7)$$

such that the differential 1-form

$$d\theta = -\frac{y-y_0}{(x-x_0)^2 + (y-y_0)^2} dx + \frac{x-x_0}{(x-x_0)^2 + (y-y_0)^2} dy, \quad (8)$$

where (x_0, y_0) is any point inside the region bounded by ψ_0 .

In our notation and interpretation of differential geometry we follow [19]. Figuratively, in any coordinate system (u, v) the 1-forms du and dv form

the lines (surfaces in higher dimensions) of constant u and v . We treat the partial derivatives ∂_u and ∂_v as vector fields, the streamlines of which give the coordinate lines of u and v . For example, if we hold v constant and vary u , we go along a streamline of ∂_u . This implies that in two dimensions du and ∂_u trace the same line. The gradients ∇u and ∇v are associated to du and dv through the metric tensor and are the vector fields that are everywhere perpendicular to the lines of constant u and v , respectively. It is important to realize that in general curvilinear coordinates ∇u and ∇v point in different directions than ∂_u and ∂_v .

2.2. Orthogonal coordinates

In general, orthogonal coordinates ζ, η with ζ aligned to ψ are described by

$$d\zeta = f(\psi)(\psi_x dx + \psi_y dy) \quad (9a)$$

$$d\eta = h(x, y)(-\psi_y dx + \psi_x dy) \quad (9b)$$

With Eq. (9) we have $g^{\zeta\eta} = \zeta_x \eta_x + \zeta_y \eta_y = 0$, $g^{\zeta\zeta} = (\nabla\psi)^2 f^2$, $g^{\eta\eta} = (\nabla\psi)^2 h^2$ and $\sqrt{g}^{-1} = (\nabla\psi)^2 h f$. From Eq. (4) we directly see that

$$\partial_\zeta = \frac{1}{(\nabla\psi)^2 f}(\psi_x \partial_x + \psi_y \partial_y) \quad (10a)$$

$$\partial_\eta = \frac{1}{(\nabla\psi)^2 h}(-\psi_y \partial_x + \psi_x \partial_y) \quad (10b)$$

i.e. ∂_ζ points into the direction of the gradient of ψ and ∂_η into the direction of constant $\psi = \text{const}$ surfaces. Note that f must be a function of ψ only since the restriction $d(d\zeta) = 0$ must hold. Furthermore, $f(\psi) = d\zeta/d\psi \neq 0$ is in principle an arbitrary function, yet we choose $f(\psi) = f_0 = \text{const}$. With this choice we directly get

$$\zeta(x, y) = f_0(\psi(x, y) - \psi_0) \quad (11)$$

Note that $\zeta_0 = 0$ and $\zeta_1 = f_0(\psi_1 - \psi_0)$. The up to now undefined function $h(x, y)$ is not arbitrary since $d(d\eta) = 0$ must hold, which we can express as

$$(\psi_x \partial_x + \psi_y \partial_y)h = -h\Delta\psi \quad (12)$$

where $\Delta\psi = \psi_{xx} + \psi_{yy}$ is again the two-dimensional Laplacian. Let us remark here that Eq. (12) can be written as $\nabla \cdot (h\nabla\psi) = 0$, which makes

the orthogonal grid an elliptic grid with adaption function h . In order to integrate this equation we need an initial condition for h . We choose to first discretize the line given by $\psi(x, y) = \psi_0$.

As already mentioned ∂_η is the vector field the streamlines of which give the coordinate lines for η . We choose $h(x, y) = \text{const}$ on ψ_0 . To this end we parameterize the coordinate line by θ

$$\left. \frac{dx}{d\theta} \right|_{\zeta=0} = \frac{x_\eta}{\theta_\eta} = \frac{-\psi_y}{\psi_x \theta_y - \psi_y \theta_x} \quad (13a)$$

$$\left. \frac{dy}{d\theta} \right|_{\zeta=0} = \frac{y_\eta}{\theta_\eta} = \frac{\psi_x}{\psi_x \theta_y - \psi_y \theta_x} \quad (13b)$$

$$\left. \frac{d\eta}{d\theta} \right|_{\zeta=0} = \frac{1}{\theta_\eta} = \frac{(\nabla\psi)^2 h(\psi_0)}{\psi_x \theta_y - \psi_y \theta_x} \quad (13c)$$

Let us define $h(\psi_0)$ such that η lies in $[0, 2\pi]$, that is,

$$2\pi = \oint_{\psi=\psi_0} d\eta = \oint_0^{2\pi} \frac{d\eta}{d\theta} d\theta$$

or

$$f_0 := h(\psi_0) = \frac{2\pi}{\int_0^{2\pi} d\theta \frac{(\nabla\psi)^2}{\psi_x \theta_y - \psi_y \theta_x}} \quad (14)$$

Here, we also fixed the constant f_0 such that our coordinate system ζ, η fulfills the Cauchy–Riemann condition on the boundary line ψ_0 . As initial point for the integration of Eq. (13) we can use any point with $\psi(x, y) = \psi_0$. We then use $h(\psi_0)$ on the flux-surface ψ_0 as initial condition for the integration of Eq. (12).

We obtain coordinate lines by integrating the vector fields ∂_ζ and ∂_η given in (10). We start the construction by integrating ∂_η for $\psi = \psi_0$, i.e. $\zeta = 0$. This can be done since $h|_{\psi_0}$ is known. The obtained points serve as starting points for the integration of $\partial_\zeta = f_0^{-1} \partial_\psi$. In order to get h we simply integrate

Eq. (12)

$$\left. \frac{dx}{d\zeta} \right|_{\eta=\text{const}} = \frac{\psi_x}{f_0(\nabla\psi)^2}, \quad (15a)$$

$$\left. \frac{dy}{d\zeta} \right|_{\eta=\text{const}} = \frac{\psi_y}{f_0(\nabla\psi)^2}, \quad (15b)$$

$$\left. \frac{dh}{d\zeta} \right|_{\eta=\text{const}} = -\frac{\Delta\psi}{f_0(\nabla\psi)^2}h. \quad (15c)$$

Note that if $\Delta\psi = 0$, we directly get a conformal grid with this algorithm. This can be seen as then $h(\zeta, \eta) = f_0$. In Appendix A we briefly study the class of functions ψ which are solutions of the Grad–Shafranov equation and satisfy $\Delta\psi = 0$. This, however, is not true in general.

Let us further remark on the sign of f_0 . It is our goal to construct a right handed coordinate system and to have $\zeta_1 > \zeta_0 = 0$. The curves of constant θ surround x_0, y_0 in a mathematically positive direction. That means that Eq. (14) implies that $f_0 > 0$ if $\nabla\psi$ points away from x_0, y_0 . If this is not the case, we obtain $f_0 < 0$. On the other hand if ζ should increase from ψ_0 to ψ_1 , we need $f_0 < 0$ for $\psi_1 < \psi_0$ and $f_0 > 0$ for $\psi_1 > \psi_0$. We thus simply take the absolute value of Eq. (14) and multiply by -1 if $\psi_1 < \psi_0$. For ease of notation we do so also in the following without further notice.

2.3. The Laplace equation

A conformal mapping has to satisfy the Cauchy–Riemann equations given in (1). A direct consequence is that u and v are harmonic functions

$$\Delta u = \Delta v = 0 \quad (16)$$

Here, $\Delta = \nabla^2$ is the two-dimensional Laplacian with the divergence and gradient operators defined in (6). Let us assume that we have constructed a flux aligned grid with coordinates (ζ, η) . This can be but, not necessarily has to be, the orthogonal coordinate system introduced in the last section. Now, in order to construct conformal coordinates u, v we first define

$$u(\zeta, \eta) := c_0(\bar{u}(\zeta, \eta) - \psi_0) \quad (17)$$

with

$$\Delta \bar{u}(\zeta, \eta) = 0 \quad (18)$$

where $\bar{u}(0, \eta) = \psi_0$ and $\bar{u}(\zeta_1, \eta) = \psi_1$ fulfills Dirichlet boundary conditions in ζ . In η we have periodic boundary conditions. We can discretize and solve Eq. (18) by any high order method that quickly converges to a solution. We choose a high order local discontinuous Galerkin method [20, 21].

Note the analogy between Eq. (17) and Eq. (11). Now, \bar{u} is equal to ψ at the boundaries and its Laplacian vanishes in between. In fact, \bar{u} takes the role of ψ in the following coordinate transformation. We introduce c_0 as a normalization constant with the same role as f_0 in the orthogonal coordinate transformation.

2.4. Conformal coordinates

For simplicity of the argument let us assume that (ζ, η) are the orthogonal coordinates from Section 2.2. We will generalize the approach in Section 2.7.

We can use $\bar{u}(\zeta, \eta)$ as well as its derivatives $\bar{u}_\zeta(\zeta, \eta)$ and $\bar{u}_\eta(\zeta, \eta)$ and Eq. (1) as a starting point to construct conformal coordinates (u, v)

$$du = c_0 (\bar{u}_\zeta d\zeta + \bar{u}_\eta d\eta) = u_x dx + u_y dy \quad (19a)$$

$$dv = c_0 \sqrt{g} (-g^{\eta\eta} \bar{u}_\eta d\zeta + g^{\zeta\zeta} \bar{u}_\zeta d\eta) = -u_y dx + u_x dy. \quad (19b)$$

The second equality is proven using the orthogonality of the (ζ, η) coordinates and the relation for the inverse derivatives Eq. (4). With this it is easy to prove that $g^{uu} = g^{vv} = (\nabla u)^2 = g^{\zeta\zeta} u_\zeta^2 + g^{\eta\eta} u_\eta^2$ and $g^{uv} = 0$. Furthermore, we get

$$\partial_u = \frac{1}{c_0 (\nabla u)^2} (g^{\zeta\zeta} u_\zeta \partial_\zeta + g^{\eta\eta} \bar{u}_\eta \partial_\eta) \quad (20a)$$

$$\partial_v = \frac{1}{c_0 \sqrt{g} (\nabla u)^2} (-\bar{u}_\eta \partial_\zeta + \bar{u}_\zeta \partial_\eta). \quad (20b)$$

As for the orthogonal coordinates we have to integrate these two vector fields to construct our coordinates. We begin with the integration of ∂_v along the $u = 0$ line. There we have

$$\partial_v|_{u=0} = \eta_v \partial_\eta = \frac{1}{c_0 \bar{u}_\zeta} \partial_\eta, \quad (21)$$

where we used $\bar{u}_\eta|_{\zeta=0} = 0$ and $\sqrt{g} g^{\zeta\zeta}|_{\zeta=0} = h(\zeta_0, \eta)/f_0 = 1$, since $h(\psi_0) = f_0$. We can use Eq. (21) to define c_0 such that $v \in [0, 2\pi]$. In order to do so we

simply integrate

$$v_1 = \int_0^{2\pi} \frac{\partial v}{\partial \eta} \Big|_{\zeta=0} d\eta = \int_0^{2\pi} \frac{1}{\eta_v} d\eta = c_0 \int_0^{2\pi} \bar{u}_\zeta(0, \eta) d\eta := 2\pi \quad (22)$$

with $v_0 = 0$ and choose c_0 such that $v_1 = 2\pi$. The integration can be done by evaluating \bar{u}_ζ at the Gaussian nodes and using Gauss–Legendre quadrature. Having done this we integrate $\partial_v \eta$ to get starting points for the integration of ∂_u from $\bar{u}_0 = 0$ to $\bar{u}_1 = c_0 \zeta_1$. In order to avoid out-of-bound errors we can artificially make the ζ, η box periodic. Note, that we can compute $u_x(\zeta, \eta)$ and $u_y(\zeta, \eta)$ from $\bar{u}(\zeta, \eta)$ by using the equations $u_x = u_\zeta \zeta_x + u_\eta \eta_x$, $u_y = u_\zeta \zeta_y + u_\eta \eta_y$ as soon as the constant c_0 becomes available.

For the integration of the vector fields we need to evaluate its components at points unequal to the Gaussian nodes. We use interpolation with the same order as the dG method we used for the solution of $\bar{u}(\zeta, \eta)$.

The result of the algorithm is the list of points $\zeta(u_i, v_i)$, $\eta(u_i, v_i)$, which we can insert into $x(\zeta, \eta)$ and $y(\zeta, \eta)$ as well as $u_x(\zeta, \eta)$ and $u_y(\zeta, \eta)$ which have been computed previously. The derivatives v_x and v_y are given by the Cauchy–Riemann equations (1).

2.5. Grid adaption

Although the conformal grid is advantageous for elliptic equations (due to the vanishing coefficients) the cell distribution is not very flexible; once the boundary is set the conformal map is unique. We therefore have little control over the distribution of cells. We can use grid adaption techniques to overcome this restriction. The idea is to modify the elliptic equations that u and v have to fulfill. That is, we solve

$$\nabla \cdot \left(\frac{\nabla u}{w} \right) = \nabla \cdot (w \nabla v) = 0 \quad (23)$$

where w is an appropriately chosen weight function. The cell size will be small in regions where w is large and spread out in regions where w is small. The Cauchy–Riemann equations (1) are changed accordingly to

$$v_x = -\frac{u_y}{w} \quad v_y = \frac{u_x}{w} \quad (24)$$

It is straightforward to implement the weight function in the orthogonal and conformal grid generation. In Section 2.2 we simply replace the function h

by h/w . Then we have

$$\partial_\zeta = \frac{1}{f_0(\nabla\psi)^2}(\psi_x\partial_x + \psi_y\partial_y) \quad (25a)$$

$$\partial_\eta = \frac{w}{h(\nabla\psi)^2}(-\psi_y\partial_x + \psi_x\partial_y) \quad (25b)$$

and $\nabla\psi \cdot \nabla(h/w) = -h/w\Delta\psi$. A suitable choice for w is

$$w = |\nabla\psi| \quad (26)$$

as then the angle-like coordinate η becomes the arc length on the ψ_0 line.

For completeness, we explicitly state the formulas for the adaption of the conformal case here. These become apparent in the next section, where we generalize this approach. In a boundary adapted grid (i.e. a grid constructed from a coordinate transform that exactly resolves the boundary) we solve the equation

$$\nabla \cdot \left(\frac{1}{w} \nabla \bar{u} \right) = 0 \quad (27)$$

with the familiar Dirichlet boundary conditions $\bar{u}|_{\partial\Omega} = \psi$. Then the coordinate lines are given by the vector fields

$$\partial_u = \frac{1}{c_0(\nabla\bar{u})^2}(g^{\zeta\zeta}\bar{u}_\zeta\partial_\zeta + g^{m\eta}\bar{u}_\eta\partial_\eta) \quad (28a)$$

$$\partial_v = \frac{w}{c_0\sqrt{g}(\nabla\bar{u})^2}(-\bar{u}_\eta\partial_\zeta + \bar{u}_\zeta\partial_\eta) \quad (28b)$$

where $(\nabla\bar{u})^2 = g^{\zeta\zeta}\bar{u}_\zeta^2 + g^{m\eta}\bar{u}_\eta^2$. The relation between u and v is given by

$$v_x = -\frac{u_y}{w}, \quad v_y = \frac{u_x}{w} \quad (29)$$

These formulas are very similar to the ones given in Section 2.4.

2.6. Monitor metric and the heat conduction tensor

We follow Reference [12, 16, 17] and replace the canonical metric tensor \mathbf{g} by a specifically tailored tensor \mathbf{G} that takes the form

$$\mathbf{G}(x, y) = \mathbf{T}\mathbf{T} + k^2\mathbf{N}\mathbf{N} + \varepsilon(x, y)\mathbf{I} \quad (30)$$

with $\mathbf{T} = (-\psi_y, \psi_x)$ and $\mathbf{N} = -(\psi_x, \psi_y)$. The vector \mathbf{T} is tangential to the contour lines of ψ while \mathbf{N} is normal to it. We have

$$\sqrt{G} = [(\varepsilon + k^2(\psi_x^2 + \psi_y^2))(\varepsilon + (\psi_x^2 + \psi_y^2))]^{-1/2} \quad (31)$$

where $k < 1$ is a constant and ε is a function that is nonzero in the neighborhood of singularities, i.e. where $\nabla\psi(x, y) = 0$. In our work we choose $k = 0.1$ and $\varepsilon(x, y) \equiv \varepsilon = 0.001$. Note that G conserves perpendicularity with respect to \mathbf{T} , i.e. if any vector $\mathbf{v} \perp \mathbf{T}$ in the canonical metric then it is also perpendicular to \mathbf{T} in \mathbf{G} . In our application this is important at the boundary.

Now, we solve

$$\begin{aligned} \nabla \cdot (\sqrt{G}\mathbf{G}\nabla u) &= 0 \\ \partial_x(\sqrt{G}(G^{xx}\partial_x u + G^{xy}\partial_y u)) + \partial_y(\sqrt{G}(G^{yx}\partial_x u + G^{yy}\partial_y u)) &= 0 \end{aligned} \quad (32)$$

with boundary conditions $u|_{\partial\Omega} = \psi$. This approach was reported to work for magnetic fields that contain singular points in the domain of interest. The resulting grid coordinate u is almost perfectly aligned in regions far away from singularities and breaks the alignment in regions where B is small or vanishes.

We now take a slightly more general approach and rewrite Eq. (32)

$$\nabla \cdot (\chi\nabla u) = 0, \quad (33)$$

where $\chi(x, y)$ is a symmetric positive-definite contravariant tensor. Then both the grid adaption from Section 2.5 as well as the monitor metric can be considered special cases. The grid adaption is recovered by setting $\chi = 1/w\mathbf{I}$, while the monitor metric is simply $\chi = \sqrt{G}\mathbf{G}$. The true conformal case is, of course, recovered by setting $\chi = \mathbf{I}$.

This allows us to provide a commonly known physical interpretation of Eq. (33). If u is a temperature, then the Dirichlet boundary condition fixes a temperature at the boundary of our domain. The tensor $\chi(x, y)$ is then the anisotropic heat conduction tensor and the coordinate lines for u are the isothermal lines of the steady state solution to the heat diffusion problem. This interpretation allows us to intuitively estimate how the coordinate lines look like when a specific χ is chosen. In the case of grid adaption, if the weight function is large the heat conduction is low resulting in small temperature gradients and thus closely spaced grid cells. On the other hand, let us

reconsider Eq. (30) for the case $\varepsilon = 0$. Then we can write

$$\chi = \frac{1}{k} \hat{t}\hat{t} + k \hat{n}\hat{n} = \chi_{\parallel} \hat{t}\hat{t} + \chi_{\perp} \hat{n}\hat{n} \quad (34)$$

which for $k < 1$ simply means that the heat conduction parallel to the magnetic field is far stronger than perpendicular to it, which is in fact the case in an actual fusion reactor. The coordinate lines will thus tend to align with the magnetic flux surfaces with the degree of alignment given by k resulting in an almost aligned grid. In the limit of vanishing k the alignment should be perfect. If the magnetic field vanishes, the tensor (30) reduces to $\chi = \mathbf{I}$.

2.7. The elliptic grids

Suppose that we have constructed a boundary aligned grid (ζ, η) , which is not necessarily orthogonal. Now, we solve the general elliptic equation

$$\nabla \cdot (\chi \nabla \bar{u}) = \partial_i (\sqrt{g} \chi^{ij} \partial_j \bar{u}) = 0 \quad (35)$$

in the transformed coordinate system. Again we set $u = c_0(\bar{u} - \psi_0)$ and require that $\bar{u}|_{\partial\Omega} = \psi$. This means that we have to transform the conduction tensor χ from Cartesian to flux coordinates, which is done by the well known rules of tensor transformation

$$\chi^{\zeta\zeta} = \zeta_x \zeta_x \chi^{xx} + 2\zeta_x \zeta_y \chi^{xy} + \zeta_y \zeta_y \chi^{yy} \quad (36a)$$

$$\chi^{\zeta\eta} = \zeta_x \eta_x \chi^{xx} + (\zeta_x \eta_y + \eta_x \zeta_y) \chi^{xy} + \zeta_y \eta_y \chi^{yy} \quad (36b)$$

$$\chi^{\eta\eta} = \eta_x \eta_x \chi^{xx} + 2\eta_x \eta_y \chi^{xy} + \eta_y \eta_y \chi^{yy}. \quad (36c)$$

The equivalent of the Cauchy–Riemann equations in this formulation reads

$$v_{\zeta} = -\sqrt{g}(\chi^{\eta\zeta} u_{\zeta} + \chi^{\eta\eta} u_{\eta}) \quad (37a)$$

$$v_{\eta} = +\sqrt{g}(\chi^{\zeta\zeta} u_{\zeta} + \chi^{\zeta\eta} u_{\eta}) \quad (37b)$$

These are constructed such that $\nabla \cdot ((\chi/\det \chi) \nabla v) = 0$. The interested reader might notice that Eq. (37) just defines the components of the Hodge dual $dv = \star du$ if χ is interpreted as a metric. We note that these equations are now valid for any grid that we use to solve Eq. (35). This specifically means that if we find a boundary aligned grid analytically we can start the grid construction directly with the solution of Eq. (35) and then proceed with the conformal grid generation.

The relevant equations for the streamline integration now read

$$du = c_0(\bar{u}_\zeta d\zeta + \bar{u}_\eta d\eta) \quad (38a)$$

$$dv = c_0\sqrt{g}(-(\chi^{\eta\zeta}\bar{u}_\zeta + \chi^{\eta\eta}\bar{u}_\eta)d\zeta + (\chi^{\zeta\zeta}\bar{u}_\zeta + \chi^{\zeta\eta}\bar{u}_\eta)d\eta) \quad (38b)$$

which just means that v is orthogonal to u in the scalar product generated by the symmetric tensor χ , which we denote by $\langle \cdot, \cdot \rangle$. We have $J = c_0^2\sqrt{g}\langle \nabla\bar{u}, \nabla\bar{u} \rangle$ and

$$\partial_u = \frac{1}{c_0\langle \nabla\bar{u}, \nabla\bar{u} \rangle}(\chi^{\zeta\zeta}\bar{u}_\zeta + \chi^{\zeta\eta}\bar{u}_\eta)\partial_\zeta + (\chi^{\eta\zeta}\bar{u}_\zeta + \chi^{\eta\eta}\bar{u}_\eta)\partial_\eta \quad (39a)$$

$$\partial_v = \frac{1}{c_0\sqrt{g}\langle \nabla\bar{u}, \nabla\bar{u} \rangle}(-\bar{u}_\eta\partial_\zeta + \bar{u}_\zeta\partial_\eta) \quad (39b)$$

The value of c_0 on ζ_0 can be chosen by using the familiar normalisation process of ∂_v on the ζ_0 line. Note that the resulting grid will, in general, not be orthogonal in the Euclidean metric. However, for all the cases we discuss in this paper, the grid is orthogonal at the boundary.

3. Numerical tests

We extend our numerical library FELTOR [www.github.com/feltor-dev/feltor](https://github.com/feltor-dev/feltor) with a geometry package that can handle the various grids discussed in Section 2. As already mentioned we use high order explicit Runge–Kutta methods (see, for example, [22]) to integrate the necessary differential equations. The grid points and metric elements are thus available up to machine precision. Note here that each point of the grids we discussed in Section 2 can be computed independently from each other, which provides a trivial parallelization option.

We perform the computationally intensive inversion of the elliptic equation (35) on a GPU with a conjugate gradient method. On a single CPU a direct solver might be preferable but the assembly of the elliptic operator is not straightforward due to the presence of the metric elements. The advantage in an iterative method is that we do not need to assemble the whole matrix we only need to implement the application to a vector.

As already discussed in the introduction in all structured grids derivatives are pulled back to a rectangular grid (the computational space). The computational space is a product space, i.e. the discretization of the derivatives is one-dimensional (cf. [23, 21]). This reduces the stencil of the discretization

compared to a discretization using an unstructured two-dimensional grid and simplifies the communication pattern in a parallel implementation. Furthermore, the matrices have a block-diagonal form with two side bands, where each block has the size $P \times P$ with P the number of polynomial coefficients in each cell. Apart from the corner entries all blocks on the diagonals are equal. This further reduces the storage requirements and in our experience increases the performance of matrix-vector multiplications by a factor of 2–3 compared to a sparse matrix format (like the compressed sparse row format) that stores all non-zero elements explicitly.

In the following we will use the function ψ that is given by Reference [24] as an analytical solution to the Grad-Shafranov equation. This function is used in practice for turbulence simulations in tokamaks. The values of the 12 coefficients in ψ can be found in Appendix B. Furthermore, all programs used for this work can be found in the latest release of FELTOR [25].

3.1. Convergence of \bar{u}

We first check that the numerical solution of Eq. (35) converges as expected. As we do not have an analytical solution we compute the difference in the $L2$ norm of the computational space between two consecutive solutions \bar{u}_{num}^1 and \bar{u}_{num}^2 . We thus define the numerical error

$$\varepsilon_{\bar{u}} = \left(\frac{\int_{\zeta_0}^{\zeta_1} d\zeta \int_0^{2\pi} d\eta (\bar{u}_{\text{num}}^1 - \bar{u}_{\text{num}}^2)^2}{\int_{\zeta_0}^{\zeta_1} d\zeta \int_0^{2\pi} d\eta (\bar{u}_{\text{num}}^1)^2} \right)^{1/2}. \quad (40)$$

In Table 1 we show the results for different high order polynomials to test the convergence of the general elliptic equation with monitor metric (35). The region is bounded by $\psi_0 = -20$ and $\psi_1 = -1$. We use an equalarc fluxgrid to discretize this region. The advantage is that this grid is quickly generated and has, other than the orthogonal grids discussed in Section 2.2, a fairly homogeneous distribution of cells throughout the domain. As we use Dirichlet boundary conditions the nonorthogonality is not an issue. We observe a quick convergence until a relative error of around 10^{-12} is reached. This apparent upper bound is due to the accuracy of the residuum in the conjugate gradient solver at 10^{-11} . Let us note that we observe similar convergence rates for the conformal and adapted grids.

3.2. Grid quality

Now we construct our elliptic grids and compare them to the near-conformal grid suggested by Reference [9] as a reference grid for existing flux aligned

$N_\eta = 10N_\zeta$	$P = 9$	$P = 11$	$P = 13$	$P = 15$
20	-	-	-	-
40	5.98E-05	8.69E-06	5.86E-06	2.87E-07
80	3.48E-06	3.40E-07	3.75E-08	1.89E-08
160	4.80E-08	3.58E-09	2.89E-10	2.64E-11
320	7.43E-10	3.02E-11	1.71E-12	3.07E-12

Table 1: Comparison of the $L2$ errors (40) for the solution of the elliptic equation with monitor metric (35) on flux grid. The number of cells is denoted by N_η and N_ζ , respectively, and we employ polynomials of degree $P - 1$ in each cell.

grids. Note that in this section by orthogonal grid we always mean the adapted orthogonal grid. In Fig. 1 we plot the orthogonal grid described in Section 2 with the function h chosen once on the inner boundary and once on the outer boundary (in both cases adaption is used), and the near conformal grid proposed by Reference [9]. In the orthogonal grids, we have an equidistant discretization at the inner respectively outer boundary. Each of these grids is aligned with the magnetic flux function ψ . The orthogonal grids distribute equally spaced cells evenly in the region around $Z = 0$, where the contour lines are almost straight. The cell size in the radial direction decreases as we go from the inner to the outer boundary. However, when the curvature of the contour lines increases (this is the case in the region around $R = 500$), the cells are either prolonged or compressed in the direction of the angle-like coordinate η . This effect results in cells with very large aspect ratio of the cells in the orthogonal grids.

The near-conformal grid is near-conformal in the sense that the conformal deformation $g^{\zeta\zeta}/g^{\eta\eta}$ is small. Note, however, that this grid is completely aligned to the function ψ . This means that the aspect ratio of the cells remains as constant as possible. The downside of this grid is that it is clearly non-orthogonal especially close to the boundary and thus Neumann boundary conditions are difficult to implement.

In Fig. 2 we plot the results from our elliptic grid generation processes. We show a true conformal grid, an elliptic grid with adaption and one constructed with a monitor metric. The last grid is clearly non-orthogonal while the conformal and adapted grids are orthogonal in the whole domain. All three

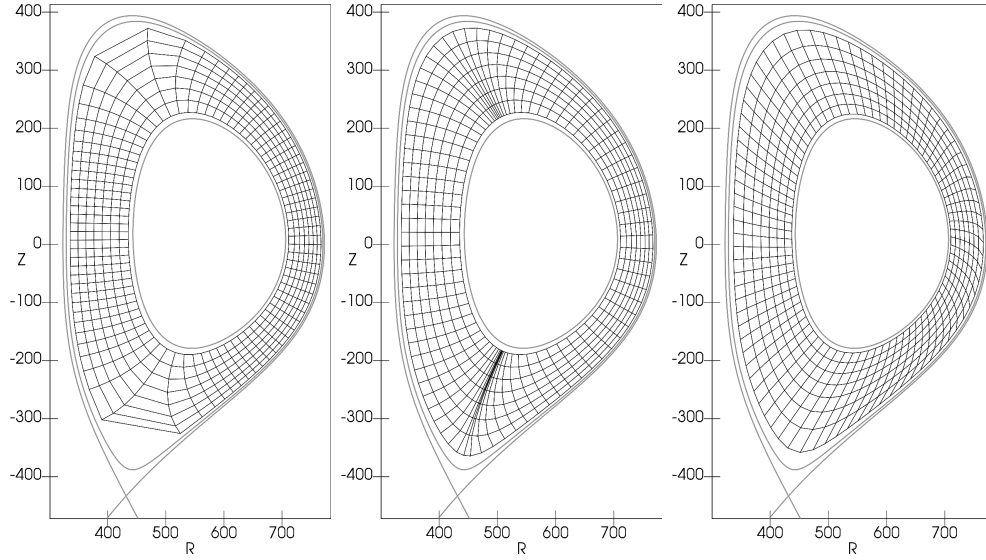


Figure 1: Orthogonal grid with $\psi_0 = -20$, $\psi_1 = -1$ (left), orthogonal grid with $\psi_0 = -1$, $\psi_1 = -20$ (middle) both with $w = |\nabla\psi|$ and near conformal grid with $\psi_0 = -20$, $\psi_1 = -1$ (right) with equal number of grid points $P = 1$, $N_\zeta = 8$, $N_\eta = 80$. Note that the nodes represent cell centers and not the actual cell boundaries. The grey lines denote the actual grid boundaries at $\psi = -20$ and $\psi = -1$. For the sake of orientation we also plot the line $\psi = 0$ to indicate the X-point.

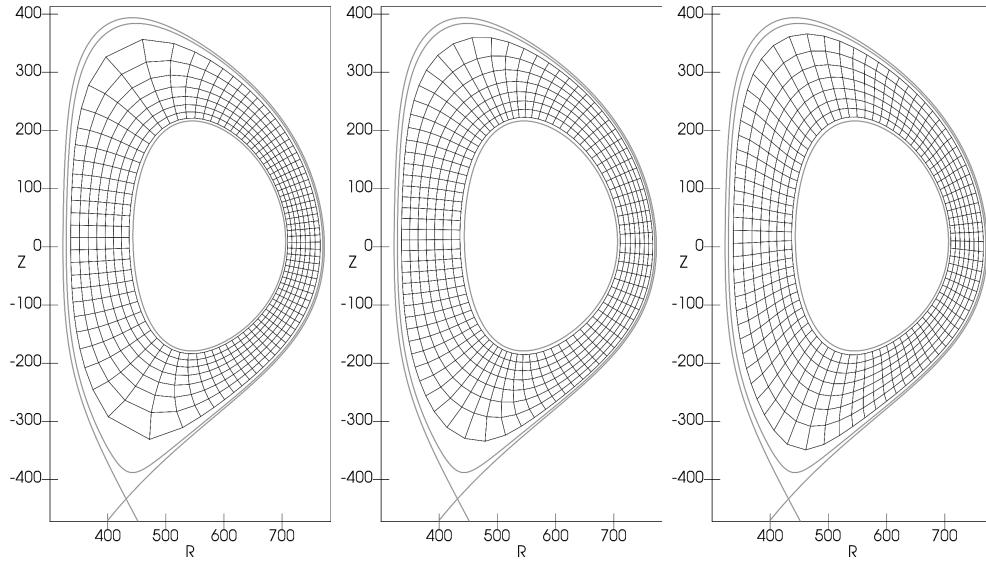


Figure 2: Truly conformal grid (left), adapted grid with $w = |\nabla\psi|$ (middle) and monitor grid with $k = 0.1$ and $\varepsilon = 0.001$ (right) with equal number of grid points $P = 1$, $N_\zeta = 8$, $N_\eta = 80$. We choose $\psi_0 = -20$ and $\psi_1 = -1$. Note that the nodes represent cell centers and not the actual cell boundaries. The grey lines denote the actual grid boundaries at $\psi = -20$ and $\psi = -1$. For the sake of orientation we also plot the line $\psi = 0$ to indicate the X-point.

grids are orthogonal at the boundary (which facilitates the implementation of Neumann boundary conditions). The conformal grid suffers from large cells in the upper and lower regions of the domain. This is cured by either performing grid adaption or by employing a monitor metric. Both result in cells of similar size. The difference between these grids is the degree of flux alignment, which is clearly superior in the monitor grid in the region close to the X-point (Reference [16] provides a more quantitative investigation of the degree of alignment).

We assess the quality of our grids using an analytic solution of the following elliptic equation

$$\nabla \cdot (\chi \nabla f) = \rho, \quad (41)$$

where we either choose a flux aligned solution

$$f_{\text{ana}}(x, y) = 0.1(\psi - \psi_0)(\psi - 2\psi_1 + \psi_0) \quad (42a)$$

$$\chi(x, y) = \frac{x_0}{x} \sqrt{1 + (\nabla \psi)^2} \quad (42b)$$

or a localized solution

$$f_{\text{ana}}(x, y) = \begin{cases} 0 & \text{if } \left(\frac{x-x_b}{\sigma}\right)^2 + \left(\frac{y-y_b}{\sigma}\right)^2 > 1 \\ \exp \left[1 + \left(\left(\frac{x-x_b}{\sigma}\right)^2 + \left(\frac{y-y_b}{\sigma}\right)^2 - 1 \right)^{-1} \right] & \text{else} \end{cases} \quad (43a)$$

$$\chi(x, y) = \frac{x_0}{x} \sqrt{1 + (\nabla \psi)^2} (1 + 0.5 \sin(\theta(x, y))). \quad (43b)$$

The corresponding parameters are as follows: $x_0 = 550$, $\psi_0 = -20$ and $\psi_1 = -1$, $x_b = 440$, $y_b = -220$ and $\sigma = 40$. The position of the blob corresponds to the lower left corner of the domain. In this region the grids differ the most. Also note that the flux aligned solution (42) has a Neumann boundary condition at $\psi = \psi_1$ and a Dirichlet boundary at $\psi = \psi_0$. For the localized solution (43) we choose Dirichlet conditions at both boundaries. In order to numerically solve Eq. (41) we first have to transform the metric and the involved functions to the coordinate system that we use (cf. Reference [23, 21] on how the derivatives are discretized). It might seem that the conformal and orthogonal grids have a performance advantage over the curvilinear grids because the non-diagonal elements of the metric vanish. However, the number of necessary matrix-vector multiplications is four in all cases and only the number of vector-vector multiplications is different. Consequently, we only observe minor differences in performance.

The relative errors are computed in the $L2$ norm via

$$\varepsilon = \left(\frac{\int_{u_0}^{u_1} du \int_0^{2\pi} dv \sqrt{g} (f_{\text{num}} - f_{\text{ana}})^2}{\int_{u_0}^{u_1} du \int_0^{2\pi} dv \sqrt{g} f_{\text{ana}}^2} \right)^{1/2} \quad (44)$$

where $\sqrt{g} du dv$ is the correct volume form in the u, v coordinate system. The ratio of N_v/N_u is chosen such that the aspect ratio of the resulting cells is approximately unity.

N_u	N_v	Conformal		Adapted		Monitor		Orthogonal	
P=3									
2	20	3.10E-02	-	1.37E-02	-	8.15E-03	-	7.34E-03	-
4	40	9.58E-03	1.69	4.80E-03	1.51	1.86E-03	2.13	4.67E-04	3.97
8	80	1.32E-03	2.86	1.17E-03	2.03	2.70E-04	2.78	4.28E-05	3.45
16	160	7.76E-04	0.76	2.74E-04	2.10	4.42E-05	2.61	4.26E-06	3.33
32	320	5.17E-05	3.91	5.02E-05	2.45	6.14E-06	2.85	3.58E-07	3.57
64	640	4.57E-06	3.50	7.41E-06	2.76	7.85E-07	2.97	3.23E-08	3.47
P=4									
2	20	5.30E-03	-	8.72E-03	-	2.72E-03	-	3.01E-04	-
4	40	1.48E-03	1.84	1.61E-03	2.43	2.21E-04	3.62	1.32E-05	4.51
8	80	9.30E-04	0.67	3.87E-04	2.06	5.47E-05	2.01	1.10E-06	3.59
16	160	1.31E-04	2.83	6.96E-05	2.48	7.57E-06	2.85	4.14E-08	4.73
32	320	4.57E-06	4.84	1.00E-05	2.80	7.83E-07	3.27	2.01E-09	4.37
64	640	5.66E-07	3.01	1.07E-06	3.22	7.93E-08	3.30	9.61E-11	4.39

Table 2: Convergence table for Eq. (41) with the flux aligned solution given in Eq. (42) on the different grids proposed. We show the relative error (44) and the corresponding order for various numbers of polynomial coefficients P in the dG discretization. The orthogonal grid uses $\psi_0 = -20$ as the first line.

Let us point out first that if we choose a function $f_{\text{ana}} = f_{\text{ana}}(\psi)$ the flux aligned grids produce errors that can be orders of magnitude lower than those of the non-aligned grids. This is observed in Table 2. The orthogonal grid has much better errors than the elliptic grids. However, while the conformal and adapted grids have similar errors the monitor grid shows smaller errors for all resolutions. This is due to the fact that the degree of alignment is higher in case of the monitor grid (see, for example [16] for a discussion). If due to some physical reasons the solution is expected to be flux aligned, the aligned

grids are thus preferable. From the dG method we expect convergence of order P [26]. This is achieved only in ideal cases, however. We observe more irregular behaviour. The orthogonal grid shows approximate orders of 3.5 and 4.5, while the remaining grids show orders mostly between 2.0 and 3.5. Since the discretization method itself is equal in all four cases this irregularity is most likely due to the different metric elements in the four coordinate systems.

In actual turbulence simulations we expect localized structures and eddies as exemplified by the localized solution given in Eq. (43). In Table 3 we show

N_u	N_v	Orthogonal		Conformal		Near Conformal		Monitor	
P=3									
4	40	1.24E+00	-	1.07E+00	-	6.67E+00	-	2.37E+00	-
8	80	3.38E-01	1.87	7.39E-01	0.54	5.35E-01	3.64	6.53E-01	1.86
16	160	2.57E-01	0.39	3.20E-01	1.21	2.92E-01	0.87	1.15E-01	2.51
32	320	3.68E-02	2.80	6.98E-02	2.20	1.76E-02	4.06	1.38E-02	3.06
64	640	3.15E-03	3.55	6.70E-04	6.70	1.58E-03	3.47	3.78E-04	5.18
P=4									
4	40	2.39E+00	-	1.49E+00	-	1.84E+00	-	1.19E+00	-
8	80	6.41E-01	1.90	8.11E-01	0.88	7.59E-01	1.28	4.09E-01	1.54
16	160	1.18E-01	2.44	1.04E-01	2.96	9.81E-02	2.95	5.56E-02	2.88
32	320	1.56E-02	2.92	2.14E-02	2.28	7.66E-03	3.68	1.93E-03	4.85
64	640	6.78E-04	4.52	3.66E-04	5.87	2.62E-04	4.87	5.15E-05	5.23

Table 3: Convergence table for the solution of Eq. (41) with the localized solution given by Eq. (43) on the different grids proposed. We show the relative error (44) and the corresponding order for two different numbers of polynomial coefficients P in the dG discretization. The orthogonal grid uses $\psi_0 = -20$ as the first line.

the resulting convergence for a forward discretization on the orthogonal grid, the truly conformal grid, the near conformal grid, and the elliptic grid with monitor metric.

Note that we again observe irregular convergence behaviour for all grids. Nonetheless we ordered the grids from mostly high to lower errors from left to right. The grid constructed with the monitor metric has the lowest error. The worst grids are the orthogonal and the true conformal grids, which have the largest cell sizes at the location, where the blob was placed. The errors from the adapted grid are approximately between the near conformal grid

and the grid with monitor metric.

The performance of these grids can be more clearly understood by comparing their maximal and minimal cell sizes for a fixed number of cells. Recall that the arc length of a curve is the integral over the norm of the tangent vector of the curve. In the case of the coordinate lines the tangent vectors are just the basis vectors ∂_u and ∂_v . If we assume that the computational space is discretized in equidistant cells with cell size h_u and h_v respectively, we define

$$l_u := \sqrt{g_{uu}}h_u = \sqrt{g}\sqrt{g^{vv}}h_u \quad (45a)$$

$$l_v := \sqrt{g_{vv}}h_v = \sqrt{g}\sqrt{g^{uu}}h_v \quad (45b)$$

Furthermore we define the ratio of maximal to minimal cell size by

$$a_u := \frac{\max l_u}{\min l_u} \quad a_v := \frac{\max l_v}{\min l_v} \quad (46)$$

	$\max l_u$	$\max l_v$	$\min l_u$	$\min l_v$	a_u	a_v
Orthogonal ($\psi_0 = -1$)	14.23	6.03	1.53	0.06	9.28	94.56
Orthogonal ($\psi_0 = -20$)	9.50	154.22	1.53	3.28	6.20	47.09
Conformal	57.02	79.51	1.75	2.44	32.53	32.53
Near Conformal	21.99	31.07	1.62	2.28	13.59	13.62
Adapted	58.91	20.05	1.68	2.04	34.98	9.82
Monitor	33.14	15.42	1.96	3.04	16.91	5.07

Table 4: Comparison of minimal and maximal cell sizes as defined in Eq. (45) and the ratios defined in Eq. (46) of the various grids with equal number of gridpoints $N_u = 32$, $N_v = 320$. The grids are ordered according to the ratio of maximum to minimum cell size in v from top to bottom.

In Table 4 we compute the maximum and minimum cell sizes of the various grids. For completeness we also computed the values for the orthogonal grid with the first line being the $\psi_0 = -1$ line, which is plotted in Fig. 1. We expect that if the maximal cell size is small, then the error committed by this grid is also small. In fact, this explains the results that were obtained in Table 3. That is, the monitor grid and the near conformal grid are better in terms of accuracy compared to the conformal and orthogonal grids. Note

that the orthogonal grid with $\psi_0 = -20$ as the first line has quite long cells in the v direction but smaller cells than all the other grids in the u direction.

The accuracy obtained is an important consideration. However, it has to be seen in the context that this elliptic equation will eventually be coupled to an evolution equation. In this case the minimal grid size is an important characteristic as it determines the CFL condition of explicit solvers and thus the computational efficiency of the grid used. Consider the minimal grid size in the v direction of the orthogonal grid with $\psi_0 = -1$ as the first line. It is 50 times smaller than the cell in the monitor grid. Thus, this grid is clearly not useable for a practical numerical simulation.

The values of a_u and a_v are a measure of how well a grid performs. Small values indicate small errors in the elliptic equation and a large minimal cell size, which is advantageous for the CFL condition. Overall we thus find that the monitor grid gives the best accuracy as well as the least stringent CFL condition among the grids we have considered here. This is expressed by a minimal value of a_v and the a_u value second only to the near conformal grid.

4. Conclusion

We show that we are able to construct various elliptic grids by the method of streamline integration combined with the solution of a suitably chosen elliptic equation. This approach can be discretized with high order Runge Kutta and dG methods. Our grids are suitable for the discretization of the edge region of magnetically confined plasmas. We show convergence of two solutions for a general elliptic equation that appears in typical physical models of plasma turbulence. Our orthogonal grids are flux aligned but show the largest variation in cell size of all the grids we investigated. The conformal grid to a lesser extent suffers from the same problem. In conclusion we find that the adaptive grid and even more the monitor metric approach yield the best grids in terms of small error and homogeneous cell distribution. We compare the errors and cell sizes to a previously suggested near conformal grid and find that our grids are competitive, which makes the orthogonality to the boundary curves the major advantage of our grids.

In the future we intend to investigate the possibility to include the X-point into the domain of interest. Furthermore, a generalization of our algorithm to three dimensions might be feasible.

Appendix A. Conformal and field aligned coordinates

The magnetic flux ψ does in general not satisfy the Laplace equation and thus is not able to serve as a conformal coordinate directly. It is therefore unclear whether it is possible to find a field aligned conformal coordinate system. Thus, we ask if it is possible to find a function u such that $u(\psi)$ satisfies the Laplace equation

$$\Delta u(\psi) = 0$$

which is equivalent to

$$(\nabla\psi)^2 u''(\psi) + (\Delta\psi) u'(\psi) = 0$$

Now we can rewrite this equation as

$$u''(\psi) = -\frac{\Delta\psi}{(\nabla\psi)^2} u'(\psi).$$

Since the left-hand side only depends on ψ , the same must hold true for the right-hand side. This implies that

$$\frac{\Delta\psi}{(\nabla\psi)^2} = g(\psi), \tag{A.1}$$

where g is an arbitrary function of ψ . Note that strictly speaking it is also possible to have $u'' = u' = 0$ which implies that $u(\psi) = \text{const.}$ This is certainly a solution but can be discarded for the purpose of constructing a coordinate system.

Equation (A.1) gives a condition on ψ that, if satisfied, allows us to construct a conformal field aligned coordinate system. Unfortunately, this is not a property that holds for the solutions of the Grad–Shafranov equation in general:

$$R \frac{\partial}{\partial R} \left(\frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial \psi}{\partial Z^2} = -\mu_0 R \frac{dp(\psi)}{d\psi} - F(\psi) \frac{dF(\psi)}{d\psi} \tag{A.2}$$

For example, $\psi(R, Z) = R^2 Z^2$ is an equilibrium solution such that

$$\frac{\Delta\psi}{(\nabla\psi)^2} = \frac{1}{2\psi} \left(1 + \frac{1}{1 + R^2/Z^2} \right)$$

which clearly does not satisfy condition (A.1).

Note, however, that for specific equilibria the condition given by equation (A.1) can be satisfied. For example, $\psi(R, Z) = R^4$ satisfies the Grad-Shafranov equation and we have

$$\frac{\Delta\psi}{(\nabla\psi)^2} = \frac{1}{\psi}.$$

Both this and the equilibrium considered above are particular cases of the class of Solovév equilibria. Although the example is uninteresting for the purpose of turbulence simulations it is noteworthy that solutions that satisfy both (A.1) and (A.2) do exist.

Appendix B. Coefficients

For ease of reproduction we print the coefficients of the solovév equilibrium solution for ψ in Reference [24]

$$\begin{aligned} A &= 0.0, \\ c_{1..12} &= [0.07350114445500399706, -0.08662417436317227513, \\ &\quad -0.14639315434011026207, -0.07631237100536276213, \\ &\quad 0.09031790113794227394, -0.09157541239018724584, \\ &\quad -0.003892282979837564482, 0.04271891225076417603, \\ &\quad 0.22755456460027913117, -0.13047241360177695448, \\ &\quad -0.03006974108476955225, 0.004212671892103931173], \\ R_0 &= 547.891714877869, \\ a/R_0 &= 0.41071428571428575, \\ \kappa &= 1.75, \\ \delta &= 0.47, \end{aligned}$$

These coefficients are also contained in the dataset for this paper [25].

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) W1227-N16 and Y398. This work has been carried out within the framework of

the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014/2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

- [1] J. Wesson, Tokamaks, Oxford University Press, 4th edition, 2011.
- [2] S. Hamada, Hydromagnetic equilibria and their proper coordinates, Nucl. Fusion 2 (1962).
- [3] A. H. Boozer, Guiding center drift equations, Phys. Fluids (1980) 904.
- [4] A. H. Boozer, Plasma equilibrium with rational magnetic surfaces, Phys. Fluids (1981) 1999.
- [5] R. Grimm, R. Dewar, M. J., Ideal MHD stability calculations in axisymmetric toroidal coordinate systems, J. Comput. Phys. 48 (1983) 94–117.
- [6] C. Z. Cheng, Kinetic extensions of magnetohydrodynamics for axisymmetric toroidal plasmas, Physics Reports 211 (1992) 1–51.
- [7] J.-k. Park, A. H. Boozer, J. E. Menard, Spectral asymmetry due to magnetic coordinates, Phys. Plasmas (2008) 064501.
- [8] O. Czarny, G. Huysmans, Bezier surfaces and finite elements for mhd simulations, J. Comput. Phys. 227 (2008) 7423–7445.
- [9] T. T. Ribeiro, B. D. Scott, Conformal tokamak geometry for turbulence computations, IEEE Transactions on Plasma Science 38 (2010) 2159–2168.
- [10] J. E. Thompson, Z. Warsi, C. W. Mastin, Numerical Grid Generation, Joe F. Thompson, 1997.
- [11] N. Weatherill, B. Soni, J. Thompson, Handbook of Grid Generation, Informa UK Limited, 1998.
- [12] V. D. Liseikin, A Computational Differential Geometry Approach to Grid Generation, Springer-Verlag, second edition, 2007.

- [13] W. D’haeseleer, W. Hitchon, J. Callen, J. Shohet, Flux Coordinates and Magnetic Field Structure, Springer Series in Computational Physics, Springer-Verlag, 1991.
- [14] J. F. Thompson, F. C. Thames, C. W. Mastin, Tomcat - code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary 2-dimensional bodies, J. Comput. Phys. 24 (1977) 274–302.
- [15] N. Papamichael, N. Stylianopoulos, Numerical Conformal Mapping, World Scientific Publishing, 2010.
- [16] A. H. Glasser, V. D. Liseikin, I. A. Vaseva, Y. V. Likhanova, Some computational aspects on generating numerical grids, Russ. J. Numer. Anal. Math. Modelling 21 (2006) 481–505.
- [17] I. A. Vaseva, V. D. Liseikin, Y. V. Likhanova, Y. N. Morokov, An elliptic method for construction of adaptive spatial grids, Russ. J. Numer. Anal. Math. Modelling 24 (2009) 65–78.
- [18] B. Scott, Shifted metric procedure for flux tube treatments of toroidal geometry: Avoiding grid deformation, Phys. Plasmas 8 (2001) 447–458.
- [19] T. Frankel, The geometry of physics: an introduction, Cambridge University Press, second edition, 2004.
- [20] B. Cockburn, G. Kanschat, I. Perugia, D. Schotzau, Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids, SIAM J. Numer. Anal. 39 (2001) 264–285.
- [21] M. Held, M. Wiesenberger, A. Stegmeir, Three discontinuous galerkin schemes for the anisotropic heat conduction equation on non-aligned grids, Comput. Phys. Commun. 199 (2016) 29–39.
- [22] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Springer-Verlag Berlin Heidelberg, 2nd edition, 1993.
- [23] L. Einkemmer, M. Wiesenberger, A conservative discontinuous galerkin scheme for the 2d incompressible navier–stokes equations, Comput. Phys. Commun. 185 (2014) 2865–2873.

- [24] A. J. Cerfon, J. P. Freidberg, "one size fits all" analytic solutions to the grad-shafranov equation, *Phys. Plasmas* 17 (2010) 032502.
- [25] M. Wiesenberger, M. Held, Feltor v3.1, 2016.
- [26] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous galerkin methods for elliptic problems, *SIAM J. Numer. Anal.* 39 (2002) 1749–1779.